Improving performance of intrusion detection system using ensemble methods and feature selection

Ngoc Tu Pham
Queensland University of Technology
Brisbane, Queensland
ngoctu.pham@connect.qut.edu.au

Ernest Foo
Queensland University of Technology
Brisbane, Queensland
e.foo@qut.edu.au

Suriadi Suriadi
Queensland University of Technology
Brisbane, Queensland
s.suriadi@qut.edu.au

Helen Jeffrey
Queensland University of Technology
Brisbane, Queensland
h.jeffrey@qut.edu.au

Hassan Fareed M Lahza
Queensland University of Technology
Brisbane, Queensland
hassan.lahza@hdr.qut.edu.au

ABSTRACT

The main task of an intrusion detection system (IDS) is to detect anomalous behaviors from both within and outside the network system, and there have been increasing studies applying machine learning in this area. The limitations of using a single classifier in the classification of normal traffic and anomalies (attacks) led to the idea of building hybrid or ensemble models which are more complicated but provide higher accuracy and lower false alarm rate (FAR). The aim of this paper is to improve the performance of IDS by using ensemble methods and feature selection. The ensemble models were built based on the two ensemble techniques, Bagging and Boosting, with the tree-based algorithms as the base classifier. The proposed models were then evaluated using NSL-KDD datasets. The experimental results showed that the bagging ensemble model with J48 as the base classifier produced the best performance in terms of both classification accuracy and FAR when working with the subset of 35 selected features.

CCS CONCEPTS

• **Security and privacy** → **Intrusion detection systems**; • **Computing methodologies** → **Ensemble methods**; **Feature selection**;

KEYWORDS

intrusion detection system, ensemble, feature selection

ACM Reference Format:
Ngoc Tu Pham, Ernest Foo, Suriadi Suriadi, Helen Jeffrey, and Hassan Fareed M Lahza. 2018. Improving performance of intrusion detection system using ensemble methods and feature selection. In *ACSW 2018: Australasian Computer Science Week 2018, January 29-February 2, 2018, Brisbane, QLD, Australia.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3167918.3167951

1 INTRODUCTION

Many approaches have been researched and developed to improve the detection rate and performance of IDSs; one of them is Machine learning (ML). ML is a ”field of study that gives computers the ability to learn without being explicitly programmed” (Arthur Samuel, 1959). ML can be applied for both signature-based and anomaly detection models. To build a signature-based detection model, all types of known attacks will be the input data in the training phase and learned by applying ML algorithms. After that, in the testing phase, each pattern in the testing dataset will be classified into classes which are different types of attacks the system learned in the training phase. In case of anomaly detection, all the features of the normal traffic in the system will be defined in the training phase, and then the built model will check each data point in the testing set as if it is normal or anomalous.

There was one problem with the initial idea of applying ML in IDS which was building a single classifier on a specific training dataset. The problem is that a single classifier may not be strong enough to build a good IDS. Thus, researchers have come up with the idea of constructing ensemble or hybrid classifiers. Many papers showed that a system using ensemble model can produce better results in classification than the one using a single classifier, such as [7][18]. A classifier ensemble was defined in [11] as ”several classifiers are employed to make a classification decision about the object submitted at the input, and the individual decisions are subsequently aggregate”. The main goal of ensemble learning is to find out the best set of classifiers and then the best method to combine them [9] instead of looking for the best feature set or building a best single classifier. There are several reasons to choose an ensemble model over a single classifier. Firstly, using an ensemble model can reduce the uncertainty in the generalization performance of using a single classifier [11]. For example, a single classifier can be trained on different subsets of a dataset and therefore produce different generalization performances. An ensemble would average the output of all these classifiers and become a better option. Secondly, when the training data is too big, it can be divided into a number of smaller sets, each set can be trained with a base classifier, and then the final outputs are aggregated, which is exactly how an ensemble model works. Finally, classifier ensemble is also useful when the training data is small. By re-sampling techniques, different datasets can be generated and trained with base classifiers in an ensemble. However, to guarantee the improvement of ensemble models over single classifiers, there are a number of factors that need to be considered, such as feature selection, base classifier and ensemble algorithms. Feature selection is important in data preprocessing because it removes redundant or irrelevant features from a given dataset to reduce the computational effort as well as improve classification accuracy. This criterion is also the most important one to choose the base classifier for an ensemble model [11]. Base classifiers are the ones whose outputs are combined by an ensemble algorithm or technique to build an ensemble model. Bagging [1] and Boosting [3] are the two most popular algorithms in ensemble learning, usually producing good results in classification and being widely chosen to build many ensemble models. This paper aims to apply these two algorithms and feature selection to build an ensemble model that can improve the performance of IDS in term of both accuracy and false alarm rate (FAR). The performance of proposed models was then evaluated on NSL-KDD datasets and compared with other existing papers.

The rest of the paper is presented as follows. Section II describes the related work, reviewing several ML-based IDSs. Section III introduces the two ensemble techniques, Bagging and Boosting, and the proposed model. Section IV presents the conducted experiments, dataset description and evaluation results. Section V discusses the provided results and finally, Section VI is dedicated to the conclusion of the paper.

2 RELATED WORK

Many IDSs have been proposed using ensemble method to improve accuracy for classification. In [19], Shrivas and Dewangan built an ensemble model based on the two classifiers Artificial Neural Network (ANN) and Bayesian Net, then combine with Gain Ratio (GR) as feature selection technique. The proposed system was evaluated on both KDD' 99 and NSL-KDD datasets and performed better than single classifiers. Meanwhile, the authors of [8] tried to reduce the number of irrelevant features in the NSL-KDD dataset by using three different algorithms for feature selection (Best First Search, Genetic, and Rank Search), and combining the results for the set of most commonly extracted features. The new dataset was then trained by three base classifiers, which are Naive Bayes, Bayesian Net and J48. For a new object, the label was gained by a majority vote from the classification result of three base classifiers. The paper showed that the proposed ensemble model was a reliable one with high true positive rate (98%) obtained from 10-fold cross validation. In another ensemble system [2], k-Nearest Neighbors and Neural Networks were used as two base classifiers in a Bagging model. The evaluated results showed the better performance of the ensemble in term of accuracy and false positive rate than any single classifier. Govindarajan [5] presented an ensemble model using Arcing (Adaptive Resampling and Combining) technique with two algorithms for base classifiers: Radial basis Function Neural Network (RBF) and Support Vector Machine (SVM). The proposed Arcing technique uses a sequential construction for base classifiers, which means the next classifier construction would depend on the performance of all previous classifiers. The classification accuracy on NSL-KDD dataset showed the outperformance of the ensemble model relative to single classifiers (RBF and SVM).

On the other hand, many hybrid approaches were produced to improve the performance of IDS. While ensemble models use ensemble techniques, such as Bagging, Boosting or Stacking, to combine base classifiers, hybrid models can use different techniques for both feature selection and building classifiers. In [16], besides decision tree and random forest, several novel approaches were applied to build a hybrid IDS, such as Principal Component Analysis (PCA), Stochastic variant of Piramol estimated sub-gradient solver in SVM (SPegasos), Ensembles of Balanced Nested Dichotomies (END), and Grading. The experimental results showed that the combination of Random Forest and END provided the highest detection rate with a low false alarm rate. Pajouh et al. introduced in [15] a two-tier network for anomaly detection based on Naive Bayes and kNN-CF classifiers. After using Linear Discriminant Analysis (LDA) for dimension reduction, the authors trained the dataset with the first classifier as Naive Bayes as the first tier, and then refine the classification using kNN-CF algorithm in the second tier. From the experimental results, Pajouh's proposed model showed improvement in both the feature reduction of the dataset and the detection of rare attack classes in classification. In [12], Malik et al. proposed the combination of Particle Swarm Optimization (PSO) and Random Forest (RF). While PSO is used to find the most relevant features for each class in the dataset, RF is used as the only classifier for classification in the model. The contribution of PSO to find more appropriate features for each class helps the proposed model produce a higher accuracy along with low false positive rate in classification compared with other algorithms.

3 PROPOSED MODEL

This section first introduced two ensemble techniques, Bagging and Boosting, which are used to combine base classifiers in proposed ensemble models. Then, the selection of tree-based algorithms as the base classifier is presented, and the last part is the design of the proposed ensemble models.

3.1 Bagging

Bagging technique [1] builds multiple classifiers based on a number of bootstrap samples from a given dataset. The label outputs of these classifiers are then decided by majority vote. The training and testing operation of Bagging algorithm is shown in Algortihm 1.

3.2 Boosting

The boosting technique used in this paper is based on AdaBoost algorithm cite, which uses distributed weights to calculate and decide the label outputs. The training and testing operation of AdaBoost algorithm is shown in Algorithm 2.

3.3 Base classifier in ensemble model

The two most important parameters of an IDS are the accuracy and false alarm rate. The accuracy shows the number of correctly detected records, while the false alarm rate demonstrates the number of benign records detected as anomalies. The selection of the base classifier in an ensemble model aims to not only increase the accuracy but also reduce the false alarm rate of the IDS. In this paper, tree-base classifiers were considered as the base learner in

```
Input: Given a labeled dataset D
Training
    (1) Choose the number of bootstrap samples n and the base classifier C;
    (2) Create n new training datasets $D_1, D_2, ..., D_n$ by sampling with replacement;
    (3) Train the base classifier on each $D_i$ to build n classifiers $C_1, C_2, ..., C_n$;
Testing
    (1) For each object x in the testing dataset, classify x by all classifiers $C_1, C_2, ..., C_n$;
    (2) Decide the label assigned to x by majority voting: the label is the class with the largest number of votes;
    (3) Repeat with all the data points in the testing set and return the outputs;
```

Algorithm 1: Bagging Algorithm

```
Input: Given a labeled dataset D with N instances
Training
    (1) Choose the base classifier C;
    (2) Set the initial weights $w_{1i} \in [0,1]$, $sum_{i=1}^{N} w_{1i} = 1$. Usually $w_{1i} = \frac{1}{\mathbf{N}}$;
    (3) For $k = 1 \longrightarrow n$, create a training sample $D_k$ from D using the distribution $w_k$;
    (4) Train the base classifier C on $D_k$ to build the classifier $C_k$;
    (5) Calculate the ensemble error $\epsilon_k = \sum_{j=1}^{N} w_{ki}$ if $C_k$ missclassifies the i_th data point in D;
    (6) If $\epsilon_k \in (0, 0.5)$, calculate $\beta_k = \frac{\epsilon_k}{1-\epsilon_k}$, and update the next weight
        $w_{k+1,i} = \begin{cases} w_{ki} \times \beta_k, & \text{if } C_k \text{ classifies correctly the i\_th data point,} \\ w_{ki} & \text{otherwise} \end{cases}$    (1)
    (7) $w_{k+1,i}$ is normalized in order to be a distribution;
    (8) For other values of $\epsilon_k$, set $w_{ki} = \frac{1}{\mathbf{N}}$ and continue;
    (9) Return the set of classifiers $C_1, C_2, ..., C_n$ and $\beta_1, \beta_2, ..., \beta_n$;
Testing
    (1) For each object x in the testing dataset, classify x by all classifiers $C_1, C_2, ..., C_n$;
    (2) For each class y assigned to x by $C_k$, calculate $\mu_y(x) = \sum_{C_k(x)=y} \ln(\frac{1}{\beta_k})$. The class with the maximum $\mu_y(x)$ is decided as the label of x;
    (3) Repeat with all the data points in the testing set and return the outputs;
```

Algorithm 2: AdaBoost Algorithm

ensemble models due to the following reasons. The first one is the sensitivity of decision tree algorithms to data resampling and feature subsampling [11], which is necessary for diversity in an ensemble model. Another reason is that decision tree classifiers can work with any type of data, both nominal and numeric features, both continuous and discrete values, and even large-scale features, so there is no need for converting, discretization or normalization. Because each split in a tree uses a best single variable, decision tree helps to reduce the use of irrelevant features in classification as well as feature selection. The tree-based classifiers selected in our experiments are the popular and widely-used algorithms, such as J48, RandomTree, REPTree, Random Forest. Based on our experiments, the difference between Bagging and Boosting techniques leads to the different selections of the base classifier for each model with each subset of features.

3.4 Proposed ensemble model

In the proposed method, both the training and testing dataset were applied feature selection in data preprocessing to remove irrelevant features. Then, a single classifier was selected as the base classifier, and the ensemble model was built by using one of the two ensemble techniques, Bagging or Boosting, to combine base classifiers. The proposed ensemble model was trained and tested with the pre-processed datasets to produce results for evaluation.

4 EXPERIMENTS

The experiments were performed by Weka 3.8.1 on Intel CORE i7-6700HQ CPU @ 2.6GHz, installed 8GB RAM, Asus laptop with 64-bit operating system. Firstly, the description of the NSL-KDD datasets [14] used in the experiments is presented. The next step is conducting the feature selection on both the NSL-KDD training and testing set. These processed datasets are used to train and test the proposed ensemble models to produce the classification accuracy and FAR. The experimental results are then evaluated and compared with the results from other existing papers.

4.1 Dataset description

NSL-KDD Dataset

Due to the limitations of the original KDD99 dataset [20], the NSL-KDD dataset was used in the experiments. The NSL-KDD dataset has been shown that it is improved than KDD99 dataset, such as removing the redundant records in the training set and duplicate records in the testing set. The number of data points in the NSL-KDD training (KDDTrain+) and testing (KDDTest+) sets are respectively 125,973 instances in the training data and 22,544 instances in the testing data. The experiments in this paper focussed on building classifier for 2-class classification: normal and anomaly (attack).

There are 41 features in NSL-KDD dataset [20], including both nominal and numeric features which are shown in Table 2.

Table 1: The number of instances in the NSL-KDD training and testing dataset

Class	KDDTrain+	KDDTest+
Normal	67343	9711
Anomaly	58630	12833
Total	125973	22544

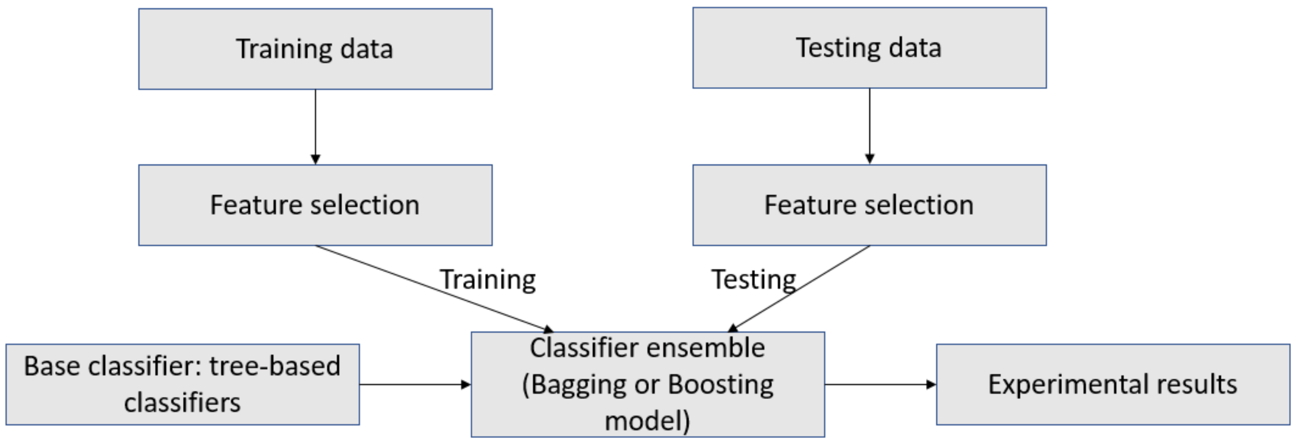

Figure 1: The proposed ensemble model

Table 2: The features in the NSL-KDD dataset

Number	Feature	Type of feature	Number	Feature	Type of feature
1	Duration	numeric	22	Is_guest_login	nominal
2	Protocol_type	nominal	23	Count	numeric
3	Service	nominal	24	Srv_count	numeric
4	Flag	nominal	25	Serror_rate	numeric
5	Src_bytes	numeric	26	Srv_serror_rate	numeric
6	Dst_bytes	numeric	27	Rerror_rate	numeric
7	Land	nominal	28	Srv_rerror_rate	numeric
8	Wrong_fragment	numeric	29	Same_srv_rate	numeric
9	Urgent	numeric	30	Diff_srv_rate	numeric
10	Hot	numeric	31	Srv_diff_host_rate	numeric
11	Num_failed_logins	numeric	32	Dst_host_count	numeric
12	Logged_in	nominal	33	Dst_host_srv_count	numeric
13	Num_compromised	numeric	34	Dst_host_same_srv_rate	numeric
14	Root_shell	numeric	35	Dst_host_diff_srv_rate	numeric
15	Su_attempted	numeric	36	Dst_host_same_src_port_rate	numeric
16	Num_root	numeric	37	Dst_host_srv_diff_host_rate	numeric
17	Num_file_creations	numeric	38	Dst_host_serror_rate	numeric
18	Num_shells	numeric	39	Dst_host_srv_serror_rate	numeric
19	Num_access_files	numeric	40	Dst_host_rerror_rate	numeric
20	Num_outbound_cmds	numeric	41	Dst_host_srv_rerror_rate	numeric
21	Is_host_login	nominal			

4.2 Feature selection

The main goal of feature selection in data preprocessing is not only to reduce the computational effort but also to find a subset that can work with the classifier to produce higher classification accuracy. Two different feature-selection techniques were applied in this paper. Firstly, based on the work of [13], 25 features were extracted from the original 41 features of NSL-KDD dataset. These 25 features were selected using "leave-one-out" techniques and Naive Bayes classifier.

The second subset consisting of 35 features was selected using Gain Ratio (GR) technique [6]. The selected features of two techniques were listed in Table 3.

4.3 Training and testing the proposed ensemble models

From the selection of the base classifier, we used two different ensemble techniques, Bagging and Boosting, to build the ensemble model, so basically, there were two different kind of ensemble models built in this paper: Bagging ensemble model and Boosting

Table 3: The selected feature from NSL-KDD dataset by 2 different techniques

Feature selection technique	Selected features	Number of features
FVBRM [13]	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 23, 24, 32, 33, 36, 38, 40	25
Gain Ratio (GR)	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17, 19, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41	35

(AdaBoost) ensemble model. These models were trained and tested respectively with the two collected subset of NSL-KDD dataset, 25-feature and 35-feature datasets. The training and testing phase were conducted on two separate datasets (KDDTrain+ and KDDTest+).

All the parameters of each algorithm used in the paper were set to default in Weka 3.8.

4.4 Experimental results

The evaluated results were based on the standard metrics for binary classification of 2 classes ”normal” and ”anomaly”: true positive (TP), false positive (FP), true negative (TN), false negative (FN)

- TP: the number of ”normal” data points classified correctly as ”normal”.
- FP: the number of ”normal” data points classified incorrectly as ”anomaly”.
- TN: the number of ”anomaly” data points classified correctly as ”anomaly”.
- FN: the number of ”anomaly” data points classified incorrectly as ”normal”.

The classification accuracy and FAR were calculated from these metrics:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

$$FAR = \frac{FP}{FP + TN} \tag{3}$$

Table 4 showed the experimental results of the proposed ensemble models trained and tested with 25-feature subset.

Table 4: Experimental results with 25-feature subset

Ensemble model	Accuracy(%)	FAR(%)
AdaBoost (Base classifier - J48)	82.40	2.88
AdaBoost (Base classifier - Random Forest)	80.58	3.90
AdaBoost (Base classifier - Decision Stump)	80.51	3.44
Bagging (Base classifier - REPTree)	**83.22**	8.09
Bagging (Base classifier - Random Tree)	81.54	3.21
Bagging (Base classifier - J48)	81.19	**2.69**

Table 5 showed the experimental results of the proposed ensemble models trained and tested with 35-feature subset.

Table 6 showed the comparisons between the bagging ensemble model with J48 as the base classifier trained and tested on 35-feature subset and other methods from existing papers

5 DISCUSSION

Experimental results with 25-feature subset:

In terms of accuracy, the best result was produced by the ensemble model using the bagging technique with REPTree as base classifier (83.22%). Two other bagging models with RandomTree and J48 as the base classifier also produced good results with accuracy (81.54% and 81.19% respectively), while the two boosting models using Random Forest and Decision Stump as base classifiers had lower numbers (80.58% and 80.51% respectively). The highest accuracy of an AdaBoost ensemble model was 82.40% when J48 was chosen as base learner.

In terms of FAR, the lower number of this parameter is, the better the intrusion detection system performs. J48-based bagging model showed the lowest number of FAR (2.69%). Despite having the best accuracy, the bagging system with REPTree as base classifier generated a much higher false alarm rate (8.09%) than other systems. All the other methods produced good performance of FAR (less than 4%).

Experimental results with 35-feature subset:

Overall, the experimental results of proposed ensemble models working with GR feature selection techniques were similar to the results of them working with 25-feature subset, except the bagging ensemble model with J48 as the base classifier. The subset of 35 selected features helped to improve greatly the classification accuracy of this ensemble model (81.19% to 84.25%) while still remaining a lower FAR than any other proposed model (2.79%). This is because GR technique tried to find the best feature with the maximum GR that can be used as a splitting note in the tree. Therefore, using GR for feature selection helped J48 algorithm overcome the bias towards features that have a large number of values [6]. This made the bagging model, with J48 as the base classifier, working on 35-feature subset have the best overall performance.

Comparisons with other approaches

To compare our proposed models with the methods from other papers, it should be noted that all the selected methods were trained and tested with NSL-KDD training and testing datasets (KDDTrain+ and KDDTest+) to produce the evaluation results of 2-class classification (normal and anomaly). The bagging model with J48 as the base classifier that worked on 35-feature subset still had better accuracy than all the proposed methods from [4][17][15]. Furthermore, the bagging ensemble model also yielded a lower FAR than any

Table 5: Experimental results with 35-feature subset

Ensemble model	Accuracy(%)	FAR(%)
AdaBoost (Base classifier - J48)	80.59	3.16
AdaBoost (Base classifier - Random Forest)	80.07	3.04
Bagging (Base classifier - REPTree)	82.61	8.14
Bagging (Base classifier - J48)	**84.25**	**2.79**

Table 6: Comparisons with other methods

Method	Accuracy(%)	FAR(%)
DAREnsemble [4]	78.88	N/A
Feature selection + SVM [17]	82.37	15
Naive Bayes – kNN-CF [15]	82	5.43
GAR-forest + Symmetrical Uncertainty [10]	85.05	12.2
Bagging (Base classifier - J48) - 35 selected features	**84.25**	**2.79**

other models in Table 6. Although the accuracy of the GAR-Forest and Symmetrical Uncertainty model in [10] was slightly better than our proposed model, the FAR of this model was much higher, which can reduce significantly the performance of an IDS. Therefore, in this paper, our ensemble models aimed to not only increase the classification accuracy but also decrease the FAR to enhance the overall performance of IDSs.

6 CONCLUSION

This paper proposed an improved IDS which used feature selection and ensemble models. The proposed models were evaluated using the NSL-KDD training and testing datasets with binary classification. Two feature-selection techniques were applied to reduce the number of irrelevant features and improve classification accuracy. The ensemble models were built based on Bagging and Boosting techniques using tree-based algorithms as the base classifier. The experimental results showed that all the proposed models had high accuracy and low FAR, and the best performance was produced by the bagging model that used J48 as the base classifier and worked on 35-feature subset (84.25% accuracy and 2.79% FAR). Although this model showed the outperformance in comparison with other existing models, there is a limitation that only one dataset was used to evaluate the built classifiers in the paper. Future work will include building classifiers for IDS working with different datasets, and improving the performance of IDS in multi-class classification using ensemble methods and feature selection.

REFERENCES

[1] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140.

[2] Shalinee Chaurasia and Anurag Jain. 2014. Ensemble neural network and k-NN classifiers for intrusion detection. *International Journal of Computer Science and Information Technology* 5 (2014), 2481–2485.

[3] Yoav Freund, Robert E Schapire, et al. 1996. Experiments with a new boosting algorithm. In *Icml*, Vol. 96. 148–156.

[4] Dwarkoba Gaikwad and Ravindra Thool. 2016. DAREnsemble: Decision Tree and Rule Learner Based Ensemble for Network Intrusion Detection System. In *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1*. Springer, 185–193.

[5] M Govindarajan. 2014. Hybrid intrusion detection using ensemble of classification methods. *International Journal of Computer Network and Information Security* 6, 2 (2014), 45.

[6] Jiawei Han, Jian Pei, and Micheline Kamber. 2011. *Data mining: concepts and techniques*. Elsevier.

[7] Lars Kai Hansen and Peter Salamon. 1990. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence* 12, 10 (1990), 993–1001.

[8] Nutan Farah Haq, Abdur Rahman Onik, and Faisal Muhammad Shah. 2015. An ensemble framework of anomaly detection using hybridized feature selection approach (HFSA). In *SAI Intelligent Systems Conference (IntelliSys), 2015*. IEEE, 989–995.

[9] Tin Kam Ho. 2002. Multiple classifier combination: Lessons and next steps. In *Hybrid methods in pattern recognition*. World Scientific, 171–198.

[10] Navaneeth Kumar Kanakarajan and Kandasamy Muniasamy. 2016. Improving the accuracy of intrusion detection using GAR-Forest with feature selection. In *Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015*. Springer, 539–547.

[11] Ludmila I Kuncheva. 2014. *Combining pattern classifiers: methods and algorithms* (second ed.). John Wiley & Sons.

[12] Arif Jamal Malik, Waseem Shahzad, and Farrukh Aslam Khan. 2015. Network intrusion detection using hybrid binary PSO and random forests algorithm. *Security and Communication Networks* 8, 16 (2015), 2646–2660.

[13] Saurabh Mukherjee and Neelam Sharma. 2012. Intrusion detection using naive Bayes classifier with feature reduction. *Procedia Technology* 4 (2012), 119–128.

[14] NSL-KDD 2017. NSL-KDD dataset. (2017). Retrieved September 20, 2017 from http://www.unb.ca/cic/research/datasets/nsl.html

[15] Hamed Haddad Pajouh, GholamHossein Dastghaibyfard, and Sattar Hashemi. 2017. Two-tier network anomaly detection model: a machine learning approach. *Journal of Intelligent Information Systems* 48, 1 (2017), 61–74.

[16] Mrutyunjaya Panda and Manas Ranjan Patra. 2009. Ensemble of classifiers for detecting network intrusion. In *Proceedings of the International Conference on Advances in Computing, Communication and Control*. ACM, 510–515.

[17] Muhammad Shakil Pervez and Dewan Md Farid. 2014. Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In *Software, Knowledge, Information Management and Applications (SKIMA), 2014 8th International Conference on*. IEEE, 1–6.

[18] Robert E Schapire. 1990. The strength of weak learnability. *Machine learning* 5, 2 (1990), 197–227.

[19] Akhilesh Kumar Shrivas and Amit Kumar Dewangan. 2014. An ensemble model for classification of attacks with feature selection based on KDD99 and NSL-KDD data set. *International Journal of Computer Applications* 99, 15 (2014).

[20] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. 2009. A detailed analysis of the KDD CUP 99 data set. In *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*. IEEE, 1–6.